

Engenharia da Computação & Bacharelado em Sistemas de Informação

Orientação a Objetos - Programação em C++

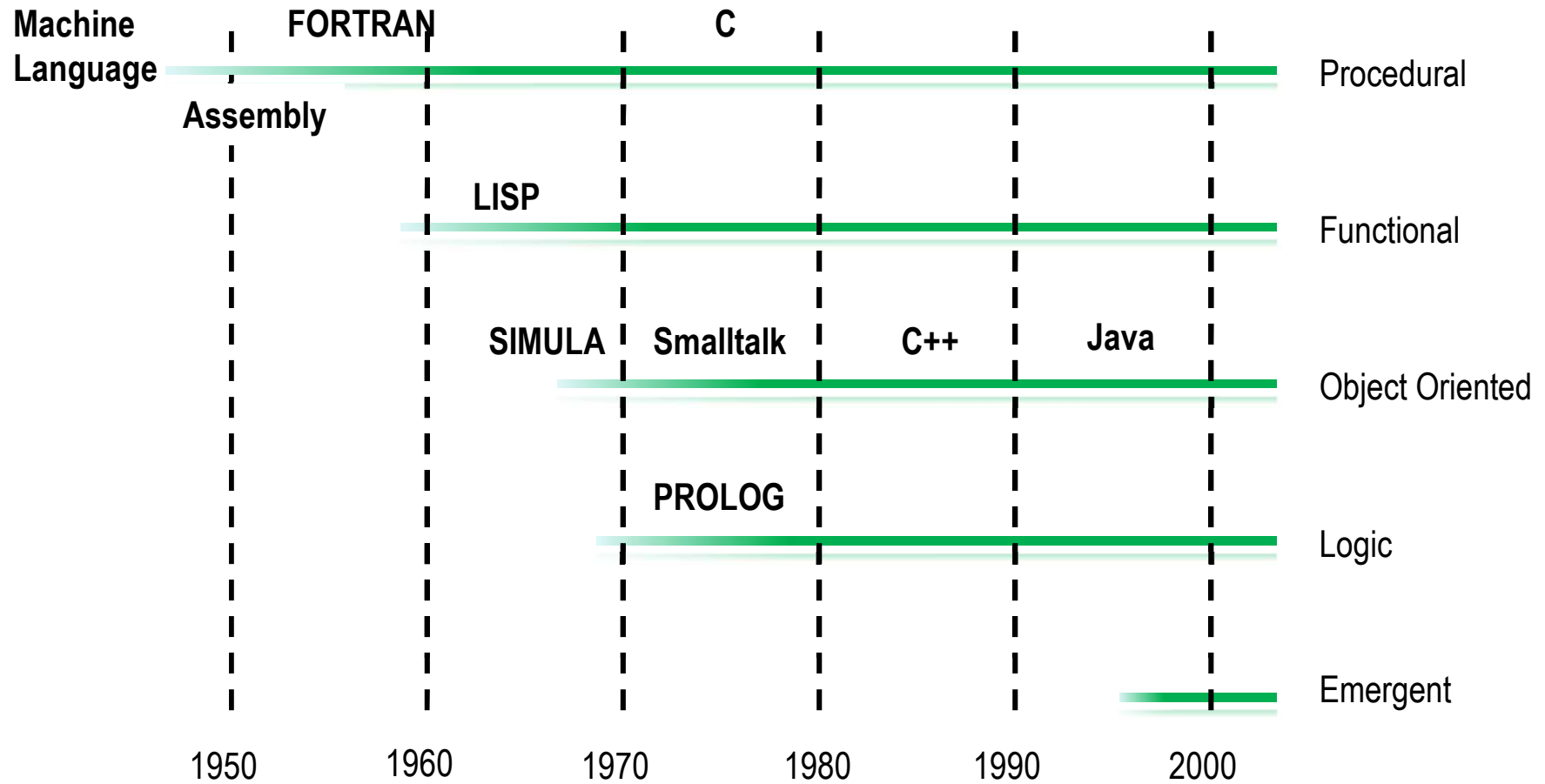
1º Conjunto de Slides – Parte A:

Introdução a Paradigmas,
Introdução à OO/C++,
Conteúdo da Disciplina e
Importância Sistêmica desta.

Prof. Jean Marcelo SIMÃO - DAINF/UTFPR

(então) Aluno Monitor: Vagner Vengue (alguns slides)

Evolução dos paradigmas



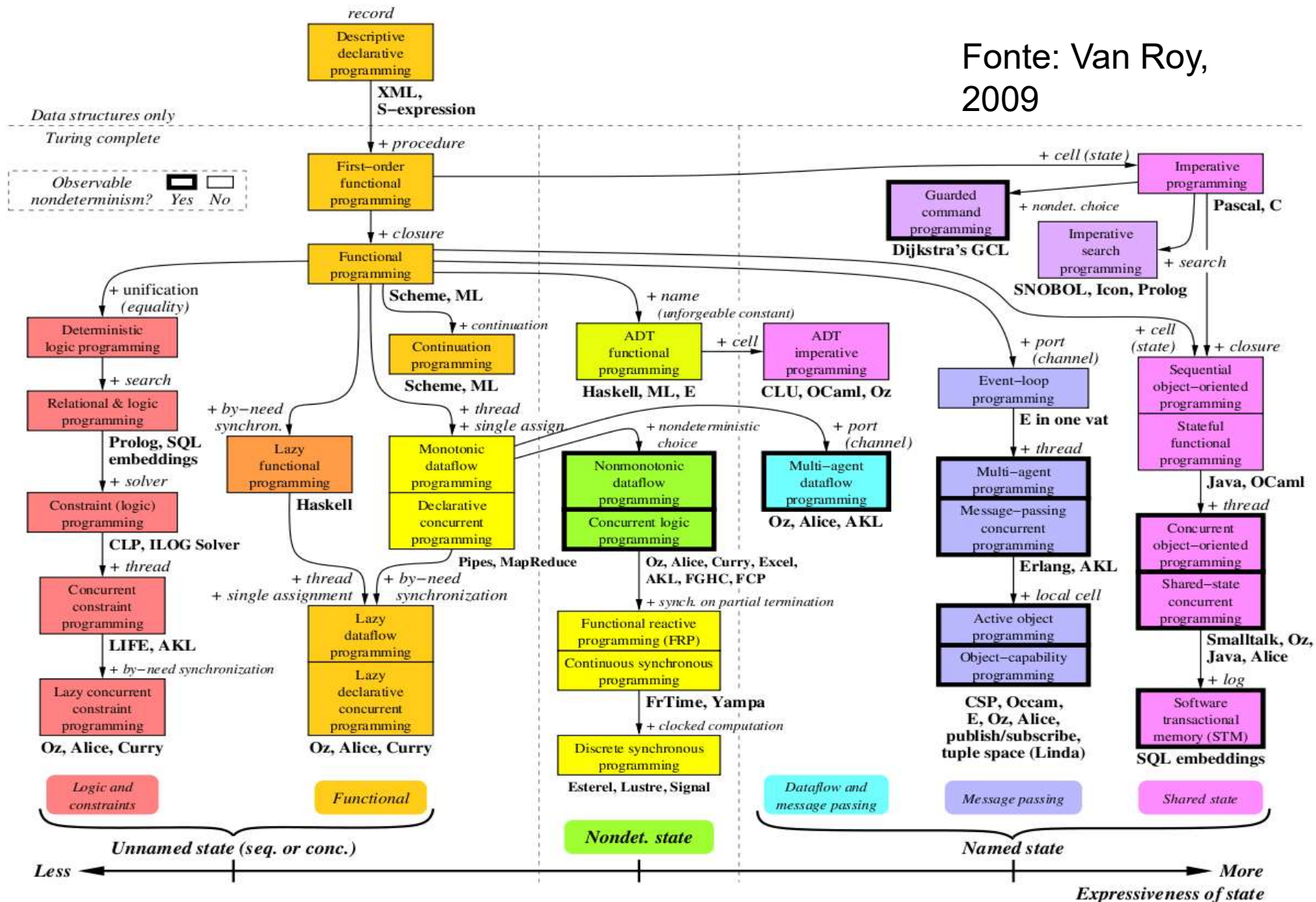
Fonte: Banaszewski, 2009

O que é um paradigma?

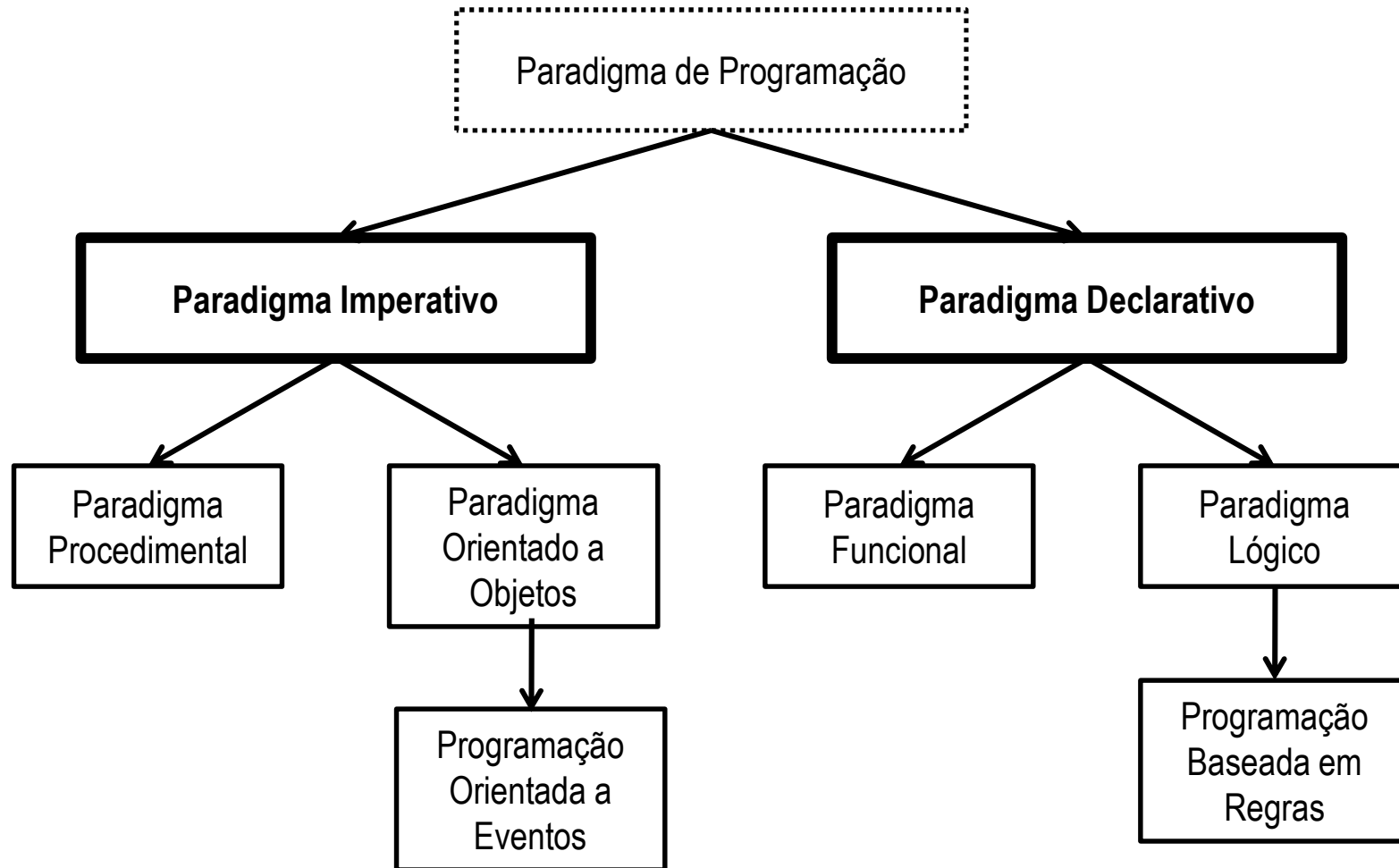
- Em ciências de forma geral
 - Segundo Thomas Kuhn:
 - Um determinado modelo que, durante um período maior ou menor de tempo e de forma mais ou menos explícita, orienta o desenvolvimento de uma pesquisa.
- Em ciência da computação
 - Segundo Peter Van Roy:
 - Um sistema formal que define como a programação é realizada. Cada paradigma possui seu próprio conjunto de técnicas e formas de estruturar o pensamento na composição de software.

Paradigmas de Programação Atuais

Fonte: Van Roy, 2009

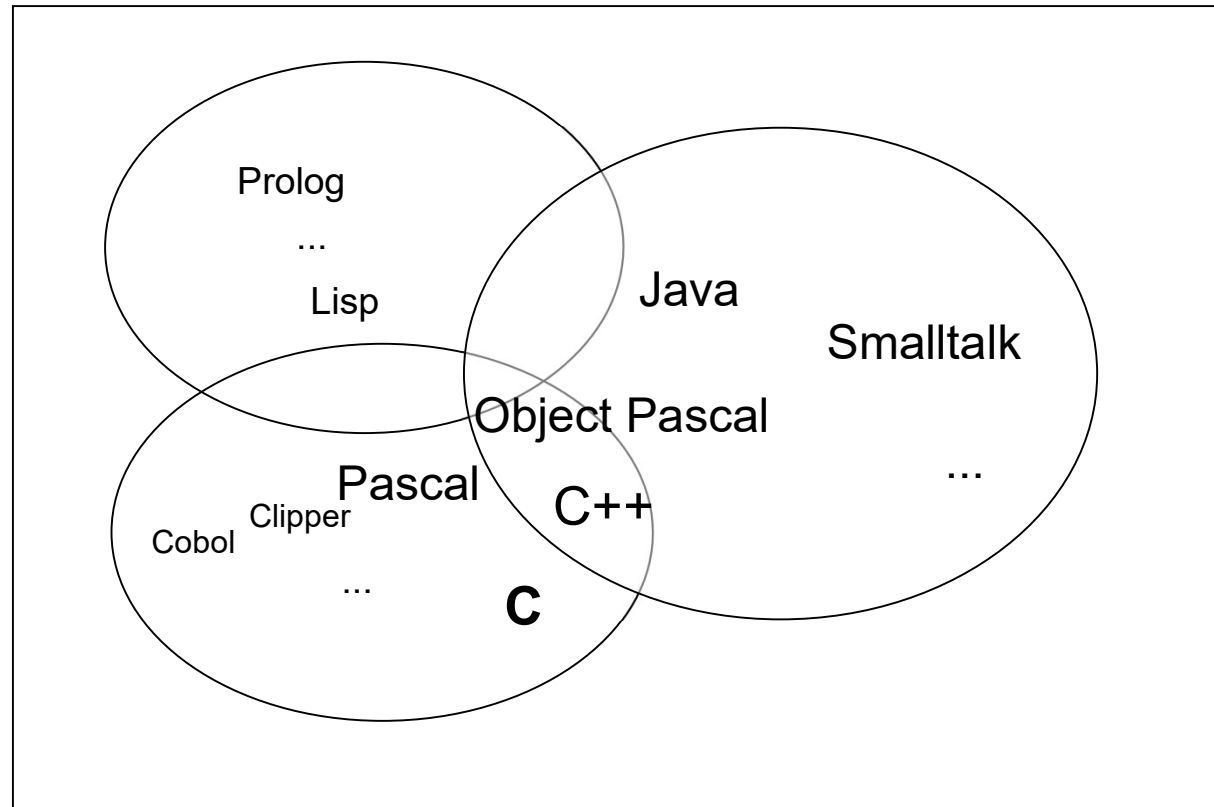


Paradigmas de Programação Atuais

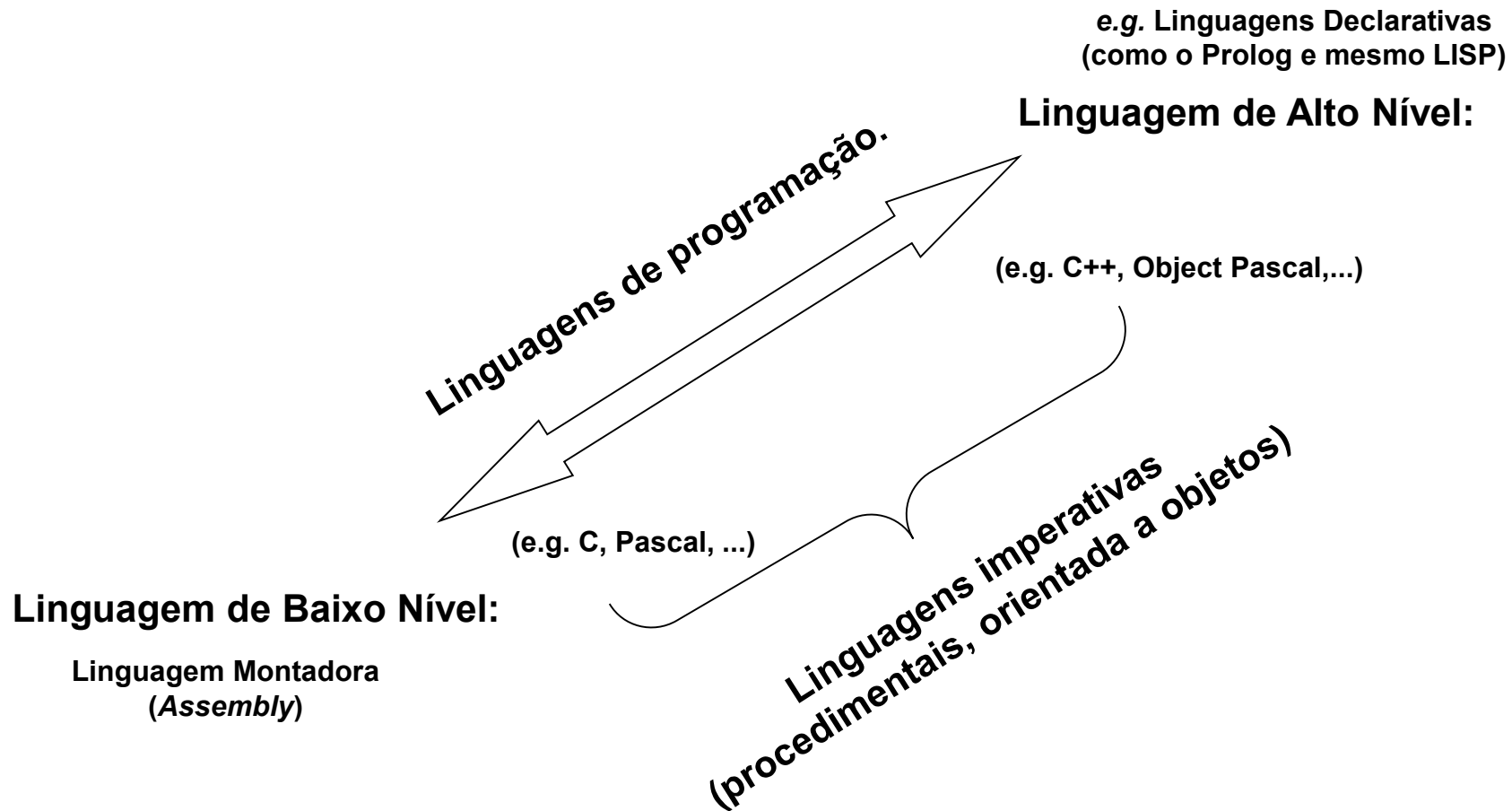


Linguagens de Programação

Uma grande diversidade de linguagens de programação



Tipos de Linguagens de Programação



Paradigmas Imperativos de Linguagem de Programação

Tipos	Procedimental	OO
Exemplos	C, Pascal, ...	C++, Object Pascal,...

Orientação a Objetos (OO)

- O que é a Orientação a Objetos (OO) ?

- É um paradigma (“modelo genérico”, conjunto de idéias) para o projeto e implementação de *softwares*, tal como o paradigma procedimental o é.

- No que concerne a implementação de *software*, existem linguagens de programação que suportam (“naturalmente”) este paradigma, sublinhando aqui o C++.

- Neste contexto, um objeto pode ser entendido como um módulo (coeso) do *software*.

OO - Vantagens

- Paradigma atual mais utilizado no desenvolvimento de novos *softwares*.
- Mais fácil de representar a solução pretendida.
 - Apresenta uma maneira mais simples de modelar o *software*, buscando-se identificar os objetos da realidade onde ele será inserido.
- Redução no custo de manutenção.
 - Quando é necessário fazer alterações no *software*, modifica-se apenas os objetos envolvidos. Como consequência, o *software* tende a ter um ciclo de vida mais longo.
- Aumento de reutilização de código.
 - Um método (i.e., uma operação) de um objeto pode ser acessado por outros objetos que o agreguem, além disso, diferentes objetos podem compartilhar os mesmos métodos.

OO - Vantagens

- Aumento de segurança de sistemas.

- Cada objeto “encapsula” o seus dados, permitindo que os outros objetos acessem apenas os seus membros (atributos e métodos) que forem permitidos. Ou seja, cada objeto deve ser responsável pelo seu estado interno.

- Melhor interação entres as fases de projeto.

- Permite a mesma abordagem desde a fase de levantamento de requisitos (ou necessidade) até a efetiva implementação.

- Técnicas mais avançadas de documentação, que permitem representar aspectos estruturais e comportamentais dos objetos.

- Torna o trabalho em equipe mais produtivo.

- Pois os integrantes não precisam conhecer todos os detalhes de implementação de cada objeto, que pode representar até subsistemas inteiros.

Origens da OO e C++

- **OO** - *Kristen Nygaard e Ole-Johan Dahl* (1962 -1967).
 - Centro Norueguês de Computação.
 - Linguagem Simula 67.
 - <http://pt.wikipedia.org/wiki/Simula>
- **OO** - Alan Kay e Equipe (~1971).
 - Xerox (Palo Alto – USA).
 - Linguagem Smalltalk.
 - <http://pt.wikipedia.org/wiki/Smalltalk>
- **C++** - Bjarne Stroustrup (1983)
 - Bell Labs (USA)
 - <http://pt.wikipedia.org/wiki/C++>

Outras referências (informais por assim dizer):

- Wikipedia http://pt.wikipedia.org/wiki/Orienta%C3%A7%C3%A3o_a_objetos
- Blog Webgoal <http://www.webgoal.com.br/desenvolvimento/origem-da-orientacao-a-objetos>

Obs.: Sítios (*sites*) visitados em 01/03/2009.

Por que C/C++ ?

Vocês aprenderam a linguagem C...

...e aprenderão C++ nesta disciplina.

De certa forma, a linguagem C++ é...

...uma evolução da linguagem C

**... que comporta (em geral) a linguagem C procedural
... bem como uma nova sintaxe orientada a objetos**

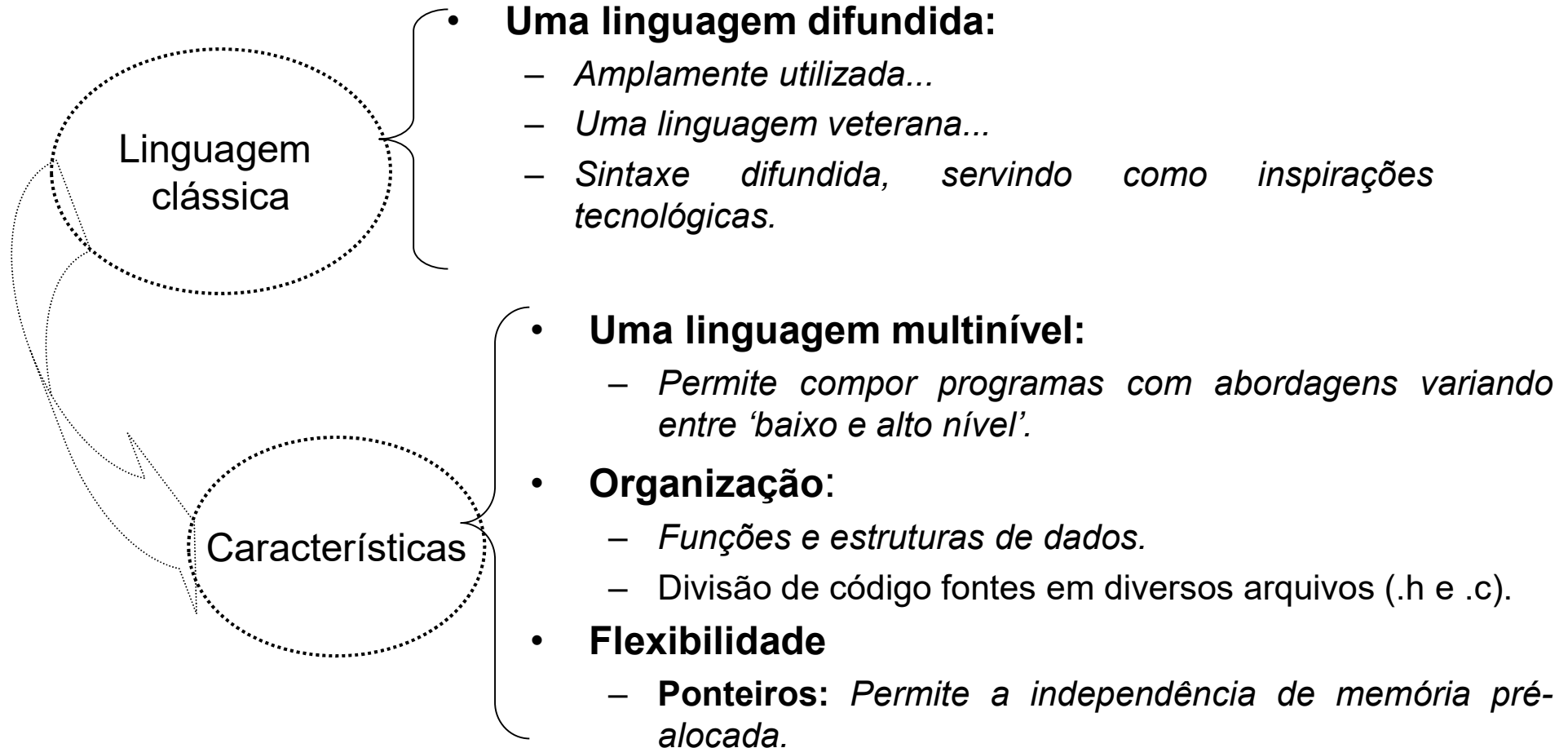


Mas...

...por que C/C++ ?

Linguagem C

[Dennis Ritchie](#) e [Ken Thompson](#), anos 70



É uma linguagem importantíssima para Eng. Eletrônica e Eng. da Computação que tendem a trabalhar com sistemas de forte intimidade eletrônico-computacional (e.g. para fins de otimizações e economias de memória) como sistemas embarcados .

Exemplo de um Código em C

```
/* Arquivo PESSOA.H */
```

```
struct Pessoa
{
    int dia;
    int mes;
    int ano;
    int idade;
};

int Calc_Idade ( struct Pessoa p, int ano );
```

```
/* Arquivo PESSOA.C */
```

```
#include "PESSOA.h"

int Calc_Idade ( struct Pessoa p, int ano )
{
    int idd = ano - p.ano;

    return idd;
}
```

```
/* Arquivo MAIN.C */
```

```
#include <stdio.h>
#include "PESSOA.h"

void main()
{
    struct Pessoa Einstein, Newton;

    Einstein.dia = 14;
    Einstein.mes = 3;
    Einstein.ano = 1879;

    Newton.dia = 4;
    Newton.mes = 1;
    Newton.ano = 1643;

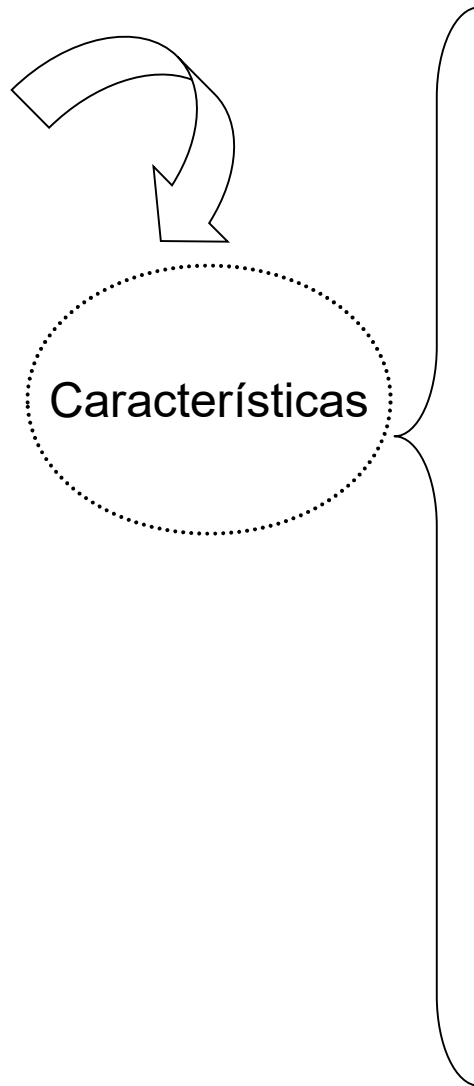
    Einstein.idade = Calc_Idade ( Einstein, 2007 );
    Newton.idade = Calc_Idade ( Newton, 2007 );

    printf("A idade de Einstein seria %d \n", Einstein.idade);
    printf("A idade de Newton seria %d \n", Newton.idade);

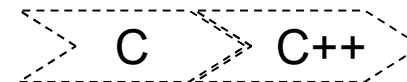
    getchar();
}
```

Exemplo de um projeto com três arquivos em Linguagem C.

Linguagem C

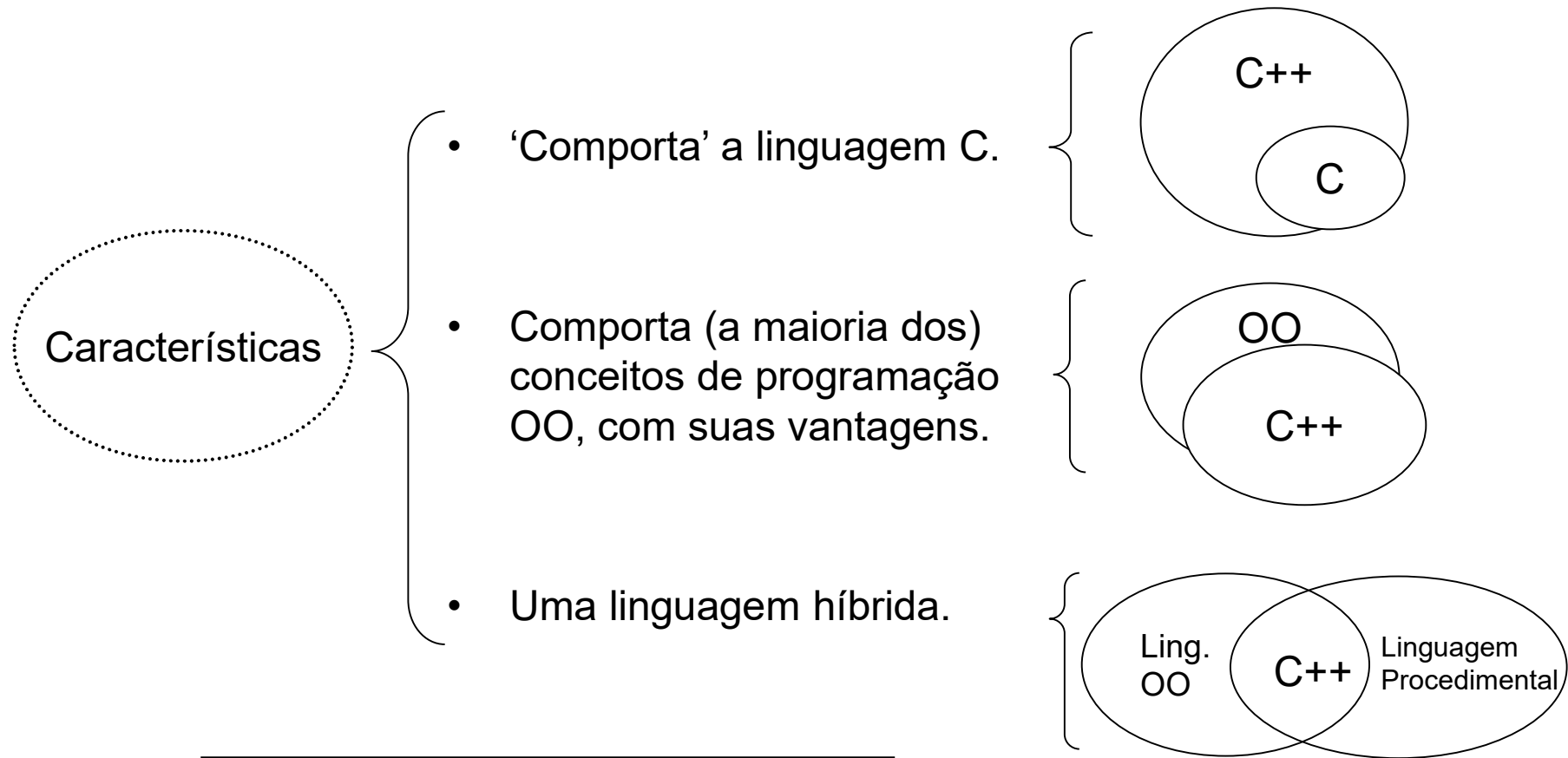


- Devido a sua **flexibilidade de utilização**, ela pode ser considerada como complicada:
 - Uma alternativa é o uso de outras linguagens “menos flexíveis”, i.e. especializada para a aplicação em questão.
 - Entretanto, o fato é que compreender C, com suas flexibilidades, permite compreender outras linguagens.
- Paradigma imperativo-procedimental:
 - O paradigma procedimental é menos rico que o paradigma orientado a objetos (OO).
 - Uma alternativa a linguagem C, seria uma linguagem OO, em particular a linguagem C++.
 - Pode-se dizer que a linguagem C++ é uma evolução da linguagem C...



Linguagem C++

Bjarne Stroustrup, anos 80



C++ : Uma linguagem flexível

Linguagem C++ e OO

Conceitos
da OO
Suportados

- Elementares:
 - Classe
 - Objetos.
- Herança.
- Agregação.
- Polimorfismo

A Orientação a Objetos (OO) permite, por exemplo, uma melhor organização e reaproveitamento de código.

Questões sistêmicas C/C++

Suporte
Tecnológico
ao C/C++

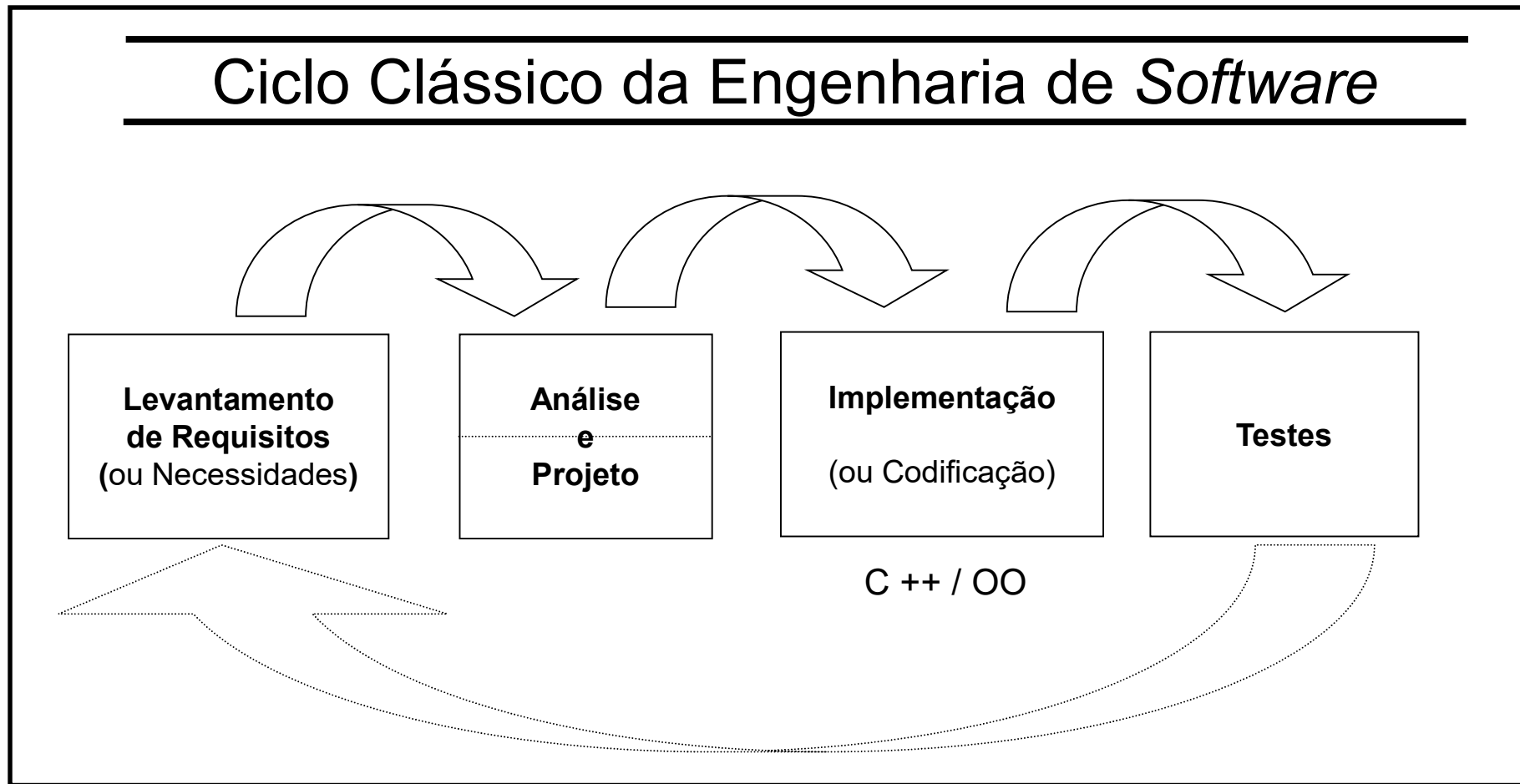
- **As linguagens C/C++ são suportada por diversos ambientes/tecnologias, por exemplo:**
 - Borland C++.
 - C++ Builder da Borland .
 - Microsoft Visual C++.
 - Microsoft Visual C++ .Net
 - Microsoft Visual C++ .Net (Express Edition)
 - G++
 - Dev C++
 - Codeblocks

Influência
Tecnológicas
do C/C++

- **Exemplos de Tecnologias Influenciadas:**
 - MatLab.
 - Java.
 - C#.

Questões sistêmicas C/C++

C++ e a Engenharia de Software



Orientação a Objetos (OO)



Questões sistêmicas C/C++

Suporte da
Engenharia
de *Software*
ao C++

- A **engenharia de software** se compõe de várias **etapas**, como análise requisitos, **projeto**, **implementação** e testes.
- **Para o projeto** existe uma Linguagem de Projeto Orientada a Objetos (OO) chamada **UML** (*Unified Modeling Language*).
- Os **conceitos da UML** são, em geral, **suportados pelo C++**.
- A maioria das ferramentas voltadas ao projeto e implementação (as **Ferramentas C.A.S.E.**) suportam UML, C++ e linguagens similares (Java e C#).
- A maioria das Ferramentas C.A.S.E (*Computer Aided Software/System Engineering*) suportam certa geração de código (em C++) a partir de diagramas UML.
- Alguns exemplos de Ferramentas C.A.S.E. são: System Architect, Mega, Rational Rose e Star UML.

Conclusão

- C/C++ constitui:
 - Uma tecnologia amplamente utilizada.
 - Uma tecnologia clássica.
 - Uma tecnologia flexível.
- C/C++ apresenta influências em desenvolvimentos tecnológicos e tem suporte tecnológico.
- C/C++ é uma tecnologia viva.
- C/C++ é uma tecnologia importante.

Conteúdos Estudados da Linguagem C

Elementos Básicos da Linguagem C/C++
- Identificadores
- Operadores Aritméticos, Lógicos e Relacionais
- Comandos de Entra/Saída
Estruturas de Decisão
- Simples (if)
- Composta (if-else)
- Múltipla (switch)
Estruturas de Repetição
- Com teste no início (while)
- Com teste no final (do – while)
- Com variável de controle (for)
Estruturas de Dados Homogêneas
- Vetores
- Matrizes
- Acesso dinâmico (ponteiros)

Estruturas de Dados Heterogêneas
- Registros de tamanho fixo (struct)
- Registros de tamanho variável (union)
- Acesso Dinâmico a estruturas
Funções / ‘Procedimentos’
- Passagem de parâmetro por valor
- Passagem de parâmetro por referência
- Retorno de valor
Arquivos
Estruturas de Dados
- Listas (encadeadas, duplamente encadeadas)
...
Recursividade
...
...
...

Conteúdo de C++ a ser estudado

Estruturas-Objetos
- Atributos
- Métodos
...
Classe-Objetos
- Atributos
- Métodos
- Construtores
- Destrutores
- Privacidade (Private, Protected, Public)
...

Relacionamentos
- Relacionamentos comuns
- Agregação
-Herança
Conceito Avançados
- Sobrecarga de método
- Sobrecarga de operador
- Classe Abstrata
- Polimorfismo
- Herança múltipla
...

Fund. 2

Dentre outros...

... vide planejamento da disciplina.

Bibliografia

C

Schildt H. "C Completo e Total". 3° ed.. Makron Books, 1996. ISBN 85-346-0595-5.

Kernighan B. W., Ritchie D. M. "The C Programming Language", 1978. Prentice-Hall. ISBN 0 -13-110163-3 (Edição Brasileira: "A linguagem de programação C". Editora Campus. 1986).

C++

Deitel H. M., Deitel, P. J. "C++ Como Programar". 3ª Edição Bookman, 2001.

Lippman S. B., Lajoie J. "C++ Primer". 3 a ed.: Addison-Wesley, 1998. ISBN 0-201-88954-4.

Stroustrup B. "C++ Programming Language". The Special Edition, Addison-Wesley (Pear), 2000. ISBN 0-201-54848-8.

Sítio (ou *site*) da Disciplina

<http://www.pessoal.utfpr.edu.br/jeansimao/Fundamentos2/Fundamentos2.htm>

<http://www.pessoal.utfpr.edu.br/jeansimao/index.htm>